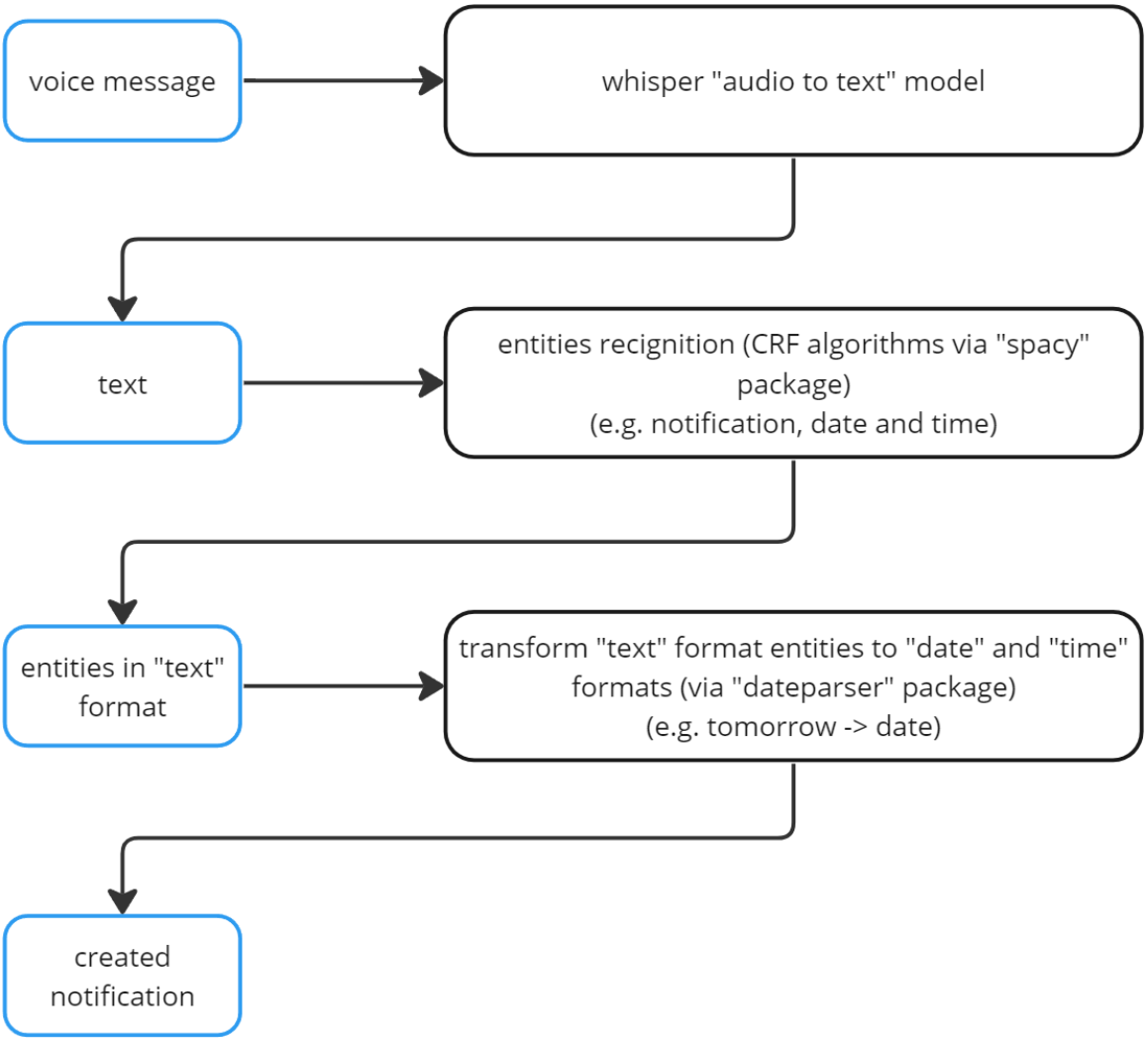
DOCUMENTATION

1) Program logic



2) How spaCy implements NER

spaCy's NER model is based on a machine learning algorithm known as a conditional random field (CRF). The model takes in a sequence of words (tokens) as input and outputs a sequence of labels indicating whether each token is part of a named entity and, if so, which type of entity it belongs to.

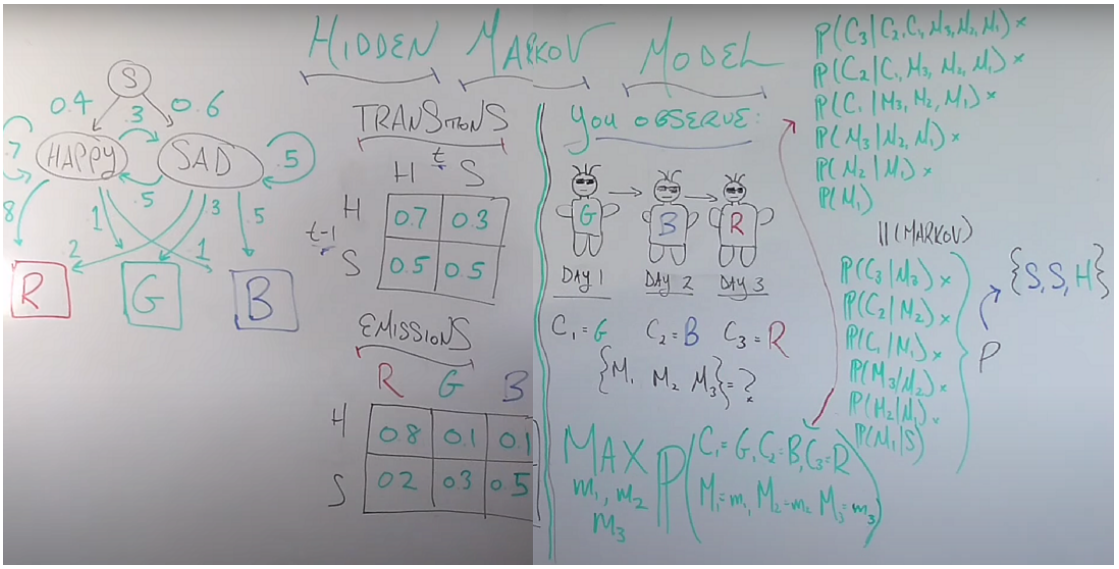
spaCy's NER model is pre-trained on a large corpus of text and can recognize a wide range of named entities out-of-the-box, including people, organizations, locations, and dates. The model also includes additional entity types, such as product names, languages, and nationalities.

3) How CRF works

Actually, CRF is the generalization of the Hidden Markov Model (HMM).

a) HMM intuitive explanation (Example)

https://www.youtube.com/watch?v=fX5bYmnHqqE&list=LL&index=34



- hidden entities: moods {happy, sad}. We should guess them by:
- t-shirt color: {R, G, B}. We can see the sequence of them (e.g. during 3 days)

- transitions matrix we get after learning
- emissions matrix we get after learning

- prediction: maximizing the target function:

$$max_{m1,m2,m3} P(C_1 = R, C_2 = G, C_3 = B, M_1 = m_1, M_2 = m_2, M_3 = m_3)$$

- if we make 2 assumptions: colors depend only on layers; current state depends only on previous (Markov condition) => we can compute the value quickly.

b) **HMM mathematical explanation**

https://en.wikipedia.org/wiki/Hidden_Markov_model

c) **CRF implementation**

https://www.youtube.com/watch?v=rI3DQS0P2fk&list=LL&index=31

Actually, everything is the same except assumptions => more complicated computations

4) Results analysis

a) Metrics (for example for "NTFY" entity:

i) TP: the number of items that marked as "NTFY" in X_test dataset and exist in y_test
ii) FP: the number of items that marked as "NTFY" in X_test dataset and do NOT exist in y_test
iii) FN: the number of items in y_test that marked as "NTFY" and do NOT exist in X_test
iv) TN the number of items in y_test that do NOT marked as "NTFY" and do NOT exist in X_test

precision = TP / (TP + FP)
recall = TP / (TP + FN)

b) Results

i) We have approximately 150 items in train dataset and 20 items in test dataset

ii) Entities metrics:

```
NTFY
precision 0.9482758620689655
recall 0.9016393442622951
TIME
precision 0.967741935483871
recall 0.9523809523809523
DATE
precision 0.9879518072289156
recall 0.9111111111111111
```

c) Progress steps

i) Problems:

NTFY: sometimes takes dates in account, do not treat "please", do not contain Geographical items

DATE: in a week is treated as "a week ago", does not understand "next week", works badly with explicit dates like "5th of…"

TIME: in 30 minutes treats as "30 min ago"

ii) Updated dataset, added new items, enlarged the number of items and the number of epochs (from 20 to 30). Problems are solved.

5) Working examples

a) Remind me today at six thirty to visit my friends
b) I plan to visit France tomorrow at 7am
c) Feed my cat the next week
d) Remind me to send a message on the April 5th, 2024 at 12:25
e) Please, check the bank account tomorrow at 3pm
f) Remind me to walk with my dog in 30 minutes

6) Notes

a) be careful while testing on your own sentences. They must have dots (.) in the end.